

# Docker编译环境使用指南

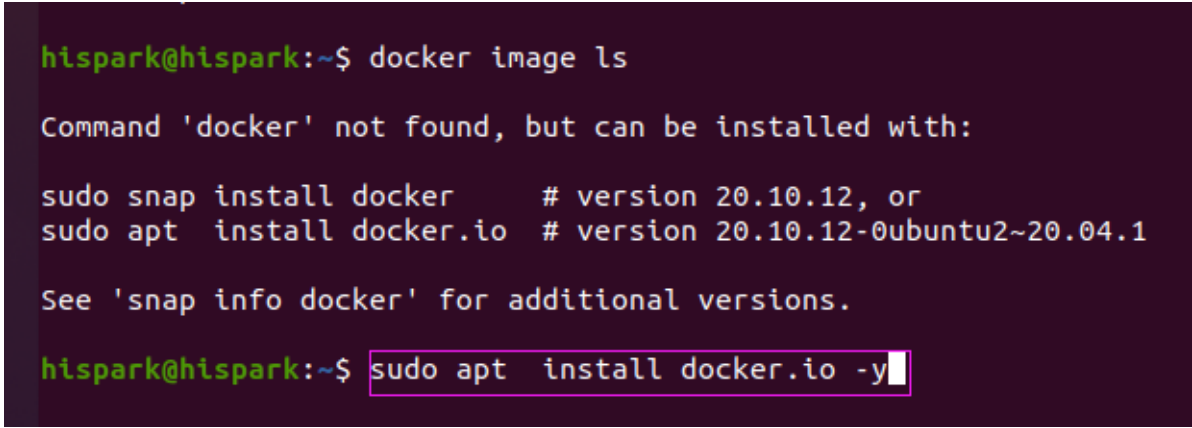
## 前提条件：

- 1、请先按照博客《[如何快速安装VMware虚拟机](#)》的步骤把虚拟机安装好。
- 2、请按照博客《[如何快速安装Ubuntu20.04](#)》的步骤把Ubuntu安装好。
- 3、如果您的Ubuntu之前下载了openharmony的代码，可以把您Ubuntu中的openharmony代码删掉，避免因为Ubuntu内存太小导致的无法预测的错误。
- 4、按照《[搭建Windows开发环境](#)》文档，搭建windows的开发工具。（已经操作过的可以忽略）
- 5、按照《[安装Remote SSH](#)》步骤，在IDE工具中安装Remote SSH插件。（已经操作过的可以忽略）

## 步骤1：安装docker软件

- 执行下面的命令，进行在Ubuntu下安装docker

```
sudo apt install docker.io -y
```



```
hispark@hispark:~$ docker image ls
Command 'docker' not found, but can be installed with:
sudo snap install docker      # version 20.10.12, or
sudo apt install docker.io    # version 20.10.12-0ubuntu2~20.04.1
See 'snap info docker' for additional versions.
hispark@hispark:~$ sudo apt install docker.io -y
```

## 步骤2：下载docker镜像包

链接: [https://pan.baidu.com/s/1czd\\_Droyrs4qP0T8c9sT7w](https://pan.baidu.com/s/1czd_Droyrs4qP0T8c9sT7w) 提取码: 17rj

## 步骤3：将docker压缩包拷贝至Ubuntu

- 关闭Ubuntu，打开编译虚拟机设置

## hispark

▶ 开启此虚拟机

✎ 编辑虚拟机设置

✎ 升级此虚拟机

## ▼ 设备

内存	8 GB
处理器	2
硬盘 (SCSI)	100 GB
CD/DVD (SATA)	正在使用未知后端
网络适配器	NAT
USB 控制器	存在
显示器	自动检测

## ▼ 描述

- 点击选项，选择共享文件夹，点击总是启用

## 虚拟机设置

硬件

选项

点击选项

设置	摘要
常规	hispark
电源	已禁用
共享文件夹	已禁用
快照	
自动保护	已禁用
客户机隔离	
访问控制	未加密
VMware Tools	关闭时间同步
VNC 连接	已禁用
Unity	
设备视图	
自动登录	不受支持
高级	默认/默认

点击共享文件夹

选择总是启用

## 文件夹共享

△ 共享文件夹会将您的文件显示给虚拟机中的程序。这可能为您的计算机和数据带来风险。请仅在您信任虚拟机使用您的数据时启用共享文件夹。

☐ 已禁用(D)☒ 总是启用(E)☐ 在下次关机或挂起前一直启用(U)

## 文件夹(E)

名称	主机路径

添加(A)...

移除(R)

属性(P)

- 点击添加按钮，然后会弹出欢迎的对话框，点击下一步



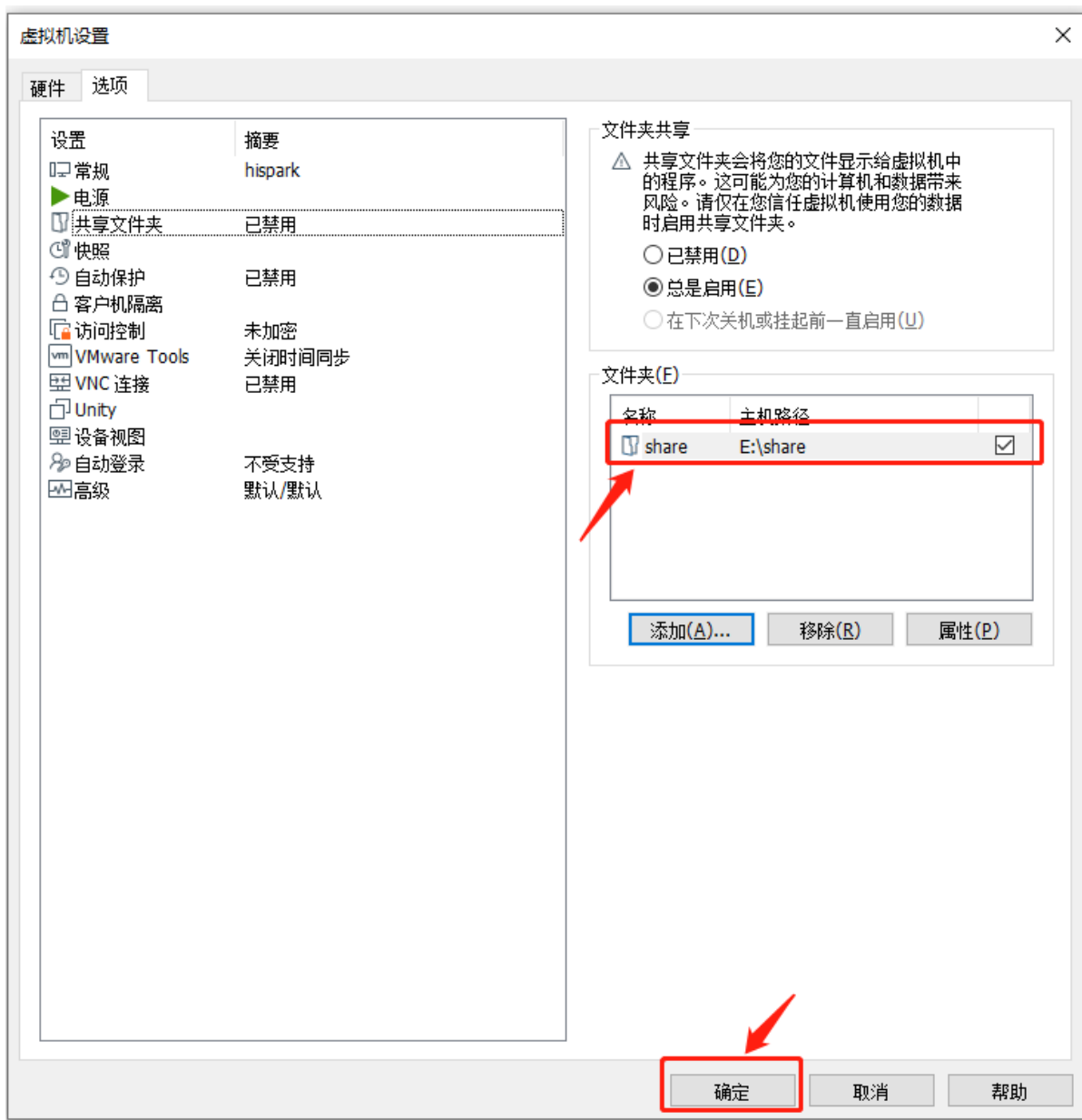
- 点击浏览按钮，选择你需要共享的Windows文件夹（后面我们把需要复制到Ubuntu的文件，放到E盘的share文件夹之后，就能在Ubuntu看到对应的文件了），然后点击下一步



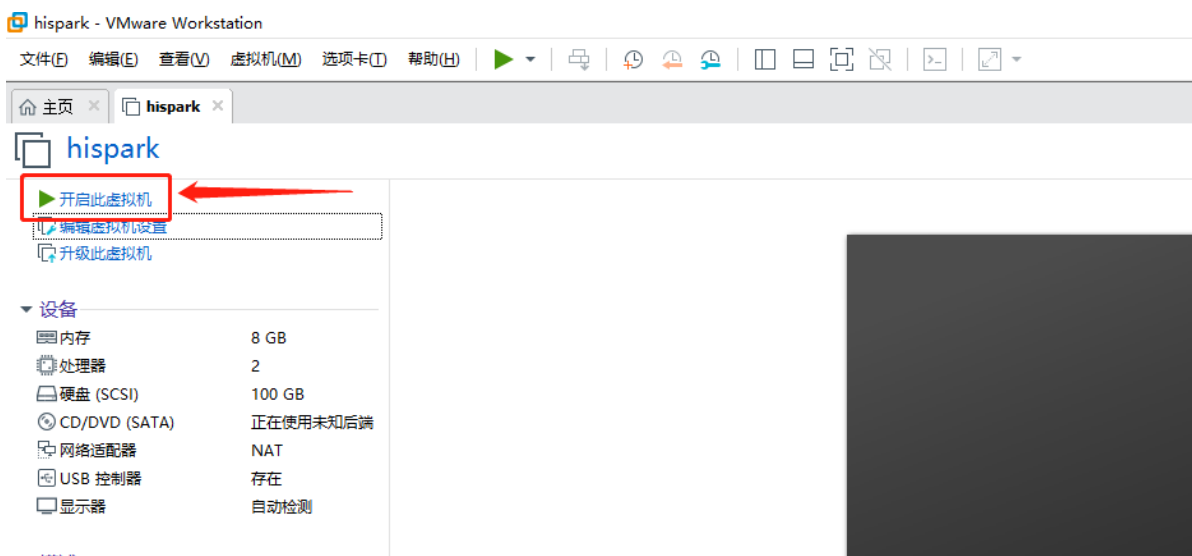
- 勾选启用此共享，然后点击完成



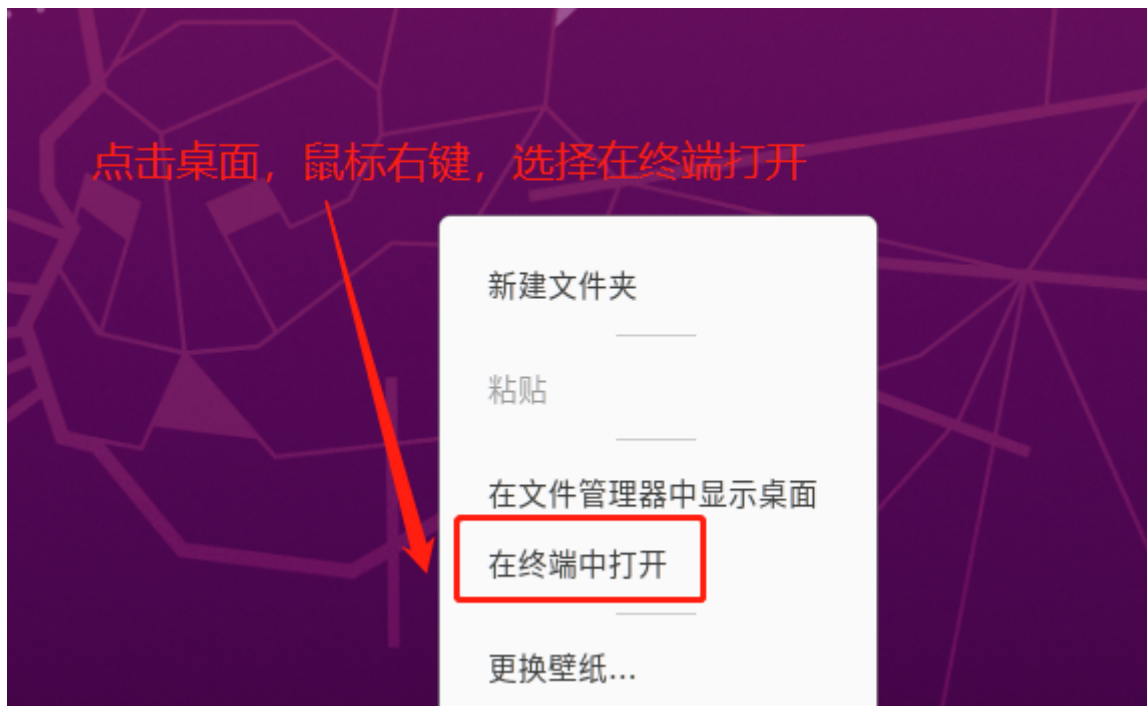
- 点击确定。



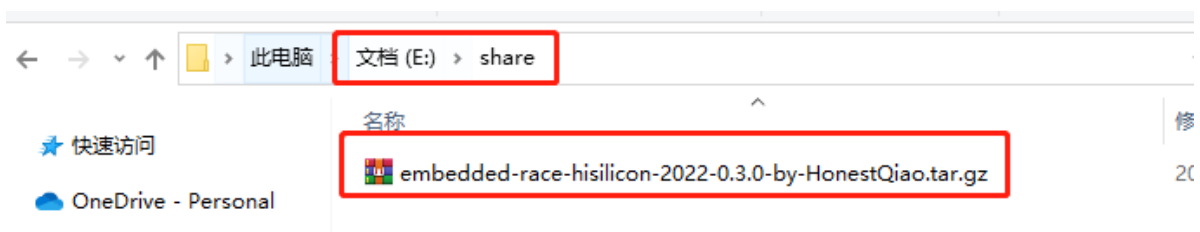
- 给Ubuntu开机。



- 等Ubuntu开机之后，输入密码，进入Ubuntu系统，在Ubuntu的桌面，鼠标右键，选择打开终端。



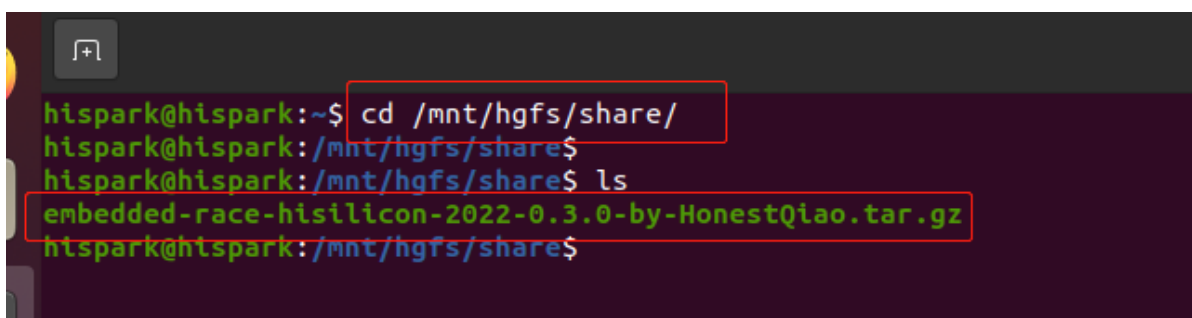
- 把下载好的docker镜像放入Windows E盘的share目录下（这个根据您自己配置的Windows的共享目录有关）



- 然后去Ubuntu的终端，执行下面的命令，就可以看到docker文件，已经在Ubuntu的share文件夹下面了。

```
cd /mnt/hgfs/share
```

```
ls
```



- 如果您的Ubuntu之前有配置过samba服务，请执行下面的命令，关闭您Ubuntu上面的samba服务，避免和后面docker镜像中使用的samba产生冲突

```
sudo service smbd stop
```

```
hispark@hispark: ~/桌面
hispark@hispark:/mnt/hgfs/share$ sudo service smbd stop
```

## 步骤4: 导入docker镜像

- 使用下面的命令, 导入docker镜像

```
sudo docker load < embedded-race-hisilicon-2022-0.3.0-by-HonestQiao.tar.gz
```

```
hispark@ubuntu:/mnt/hgfs/share$ sudo docker load < embedded-race-hisilicon-2022-0.3.0-by-HonestQiao.tar.gz
9f54eef41275: Loading layer [====>] 75.16MB/75.16MB
d5dad01a22f5: Loading layer [====>] 7.168kB/7.168kB
c38ca077ff39: Loading layer [====>] 927.7kB/927.7kB
506e35b235f3: Loading layer [====>] 35.18MB/35.18MB
b9db469aab68: Loading layer [====>] 35.86MB/35.86MB
03a830d49dd5: Loading layer [====>] 1.233GB/1.233GB
b766bbf0f3ec: Loading layer [====>] 10.75kB/10.75kB
91ec5c200fb3: Loading layer [====>] 46.47MB/46.47MB
997cdc87d415: Loading layer [====>] 83.88MB/83.88MB
3a494a4a2848: Loading layer [====>] 2.45MB/2.45MB
535f4ee8a26e: Loading layer [====>] 7.817MB/7.817MB
397cb67db14e: Loading layer [====>] 230.1MB/230.1MB
8ac0d2ad4c41: Loading layer [====>] 754.4MB/754.4MB
32f7bb7f262b: Loading layer [====>] 4.637GB/4.637GB
13feb4172acc: Loading layer [====>] 9.693GB/9.693GB
5f70bf18a086: Loading layer [====>] 1.024kB/1.024kB
a047118371b9: Loading layer [====>] 5.703GB/5.703GB
Loaded image: embedded-race-hisilicon-2022:0.3.1
hispark@ubuntu:/mnt/hgfs/share$
```

## 步骤5: 查看docker image信息

- 使用下面的命令, 查看docker image的具体信息

```
sudo docker image ls
```

```
hispark@ubuntu:/mnt/hgfs/share$ sudo docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
embedded-race-hisilicon-2022  0.3.1      7f917f0e7a6f  2 hours ago  22.2GB
hispark@ubuntu:/mnt/hgfs/share$
```

## 步骤6: 启动docker编译环境

- 执行下面的命令, 启动docker编译环境, 其中openharmony为自定义的docker的名字, 7f917f0e7a6f为我Ubuntu下docker的IMAGE ID, 这里请根据自己docker image ID的不同自行修改。

```
sudo docker run -it --net=host --name openharmony -v
/home/hisparm/docke_r_share:/home/share 7f917f0e7a6f

# sudo docker run -it --net=host --name [容器实例名] -v /home/你Ubuntu的用户
名/docke_r_share:/home/share [镜像ID]
```

```
hispark@ubuntu:/mnt/hgfs/share$ sudo docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
embedded-race-hisilicon-2022  0.3.1      7f917f0e7a6f  2 hours ago  22.2GB
hispark@ubuntu:/mnt/hgfs/share$ sudo docker run -it --net=host --name openharmony -v /home/hisparm/docke_r_share:/home/share 7f917f0e7a6f
* Restarting OpenBSD Secure Shell server sshd
root@ubuntu:/home/openharmony#
```

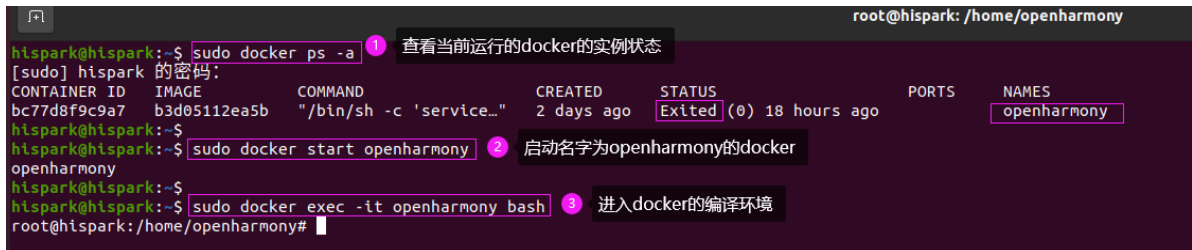
- 如果您的Ubuntu重启之后, 需要再次启动dcoker编译环境的话, 需要执行下面的命令

```
# 查看当前运行的docker实例状态
sudo docker ps -a

# 在上一条指显示结果列表中，查看openharmy的STATUS
# 如为 Exited，则需要执行下面这条指令，再次启动
# 如为 Up，则跳过下面这条指令
sudo docker start openharmony

# 进入docker编译环境
sudo docker exec -it openharmony bash

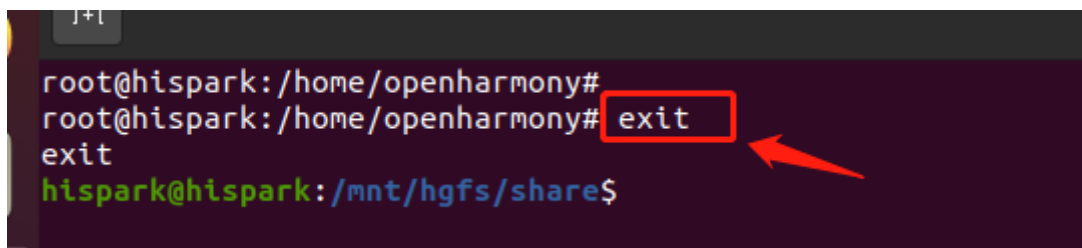
# 执行后，出现类似如下信息，说明再次进入成功
# root@bae85ba0f77c:/home/openharmony#
```



```
hispark@hispark:~$ sudo docker ps -a
[sudo] hispark 的密码:
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS      PORTS   NAMES
bc77d8f9c9a7   b3d05112ea5b   "/bin/sh -c 'service..." 2 days ago    Exited(0)   18 hours ago
hispark@hispark:~$ sudo docker start openharmony
openharmony
hispark@hispark:~$ sudo docker exec -it openharmony bash
root@hispark:/home/openharmony#
```

- 如果你想退出docker的编译环境，需要执行下面的命令

```
exit
```

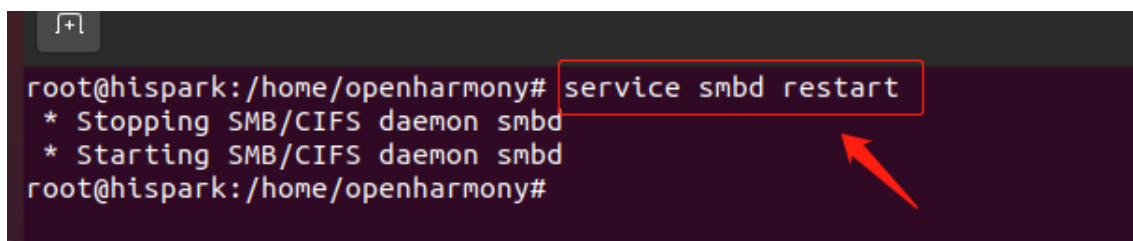


```
root@hispark:/home/openharmony#
root@hispark:/home/openharmony# exit
exit
hispark@hispark:/mnt/hgfs/share$
```

## 步骤7：配置docker的samba服务，方便Windows与docker之间的文件共享

- 输入下面的命令，启动docker的samba服务

```
service smb restart
```



```
root@hispark:/home/openharmony# service smb restart
* Stopping SMB/CIFS daemon smb
* Starting SMB/CIFS daemon smb
root@hispark:/home/openharmony#
```

- 执行下面的命令，查看Ubuntu的IP地址，本人Ubuntu的IP地址是 192.168.174.168

```
ifconfig
```



```

root@hispark:/home/openharmony# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:93:0f:8a:4a txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.174.168 netmask 255.255.255.0 broadcast 192.168.174.255
    inet6 fe80::7633:a87c:2e2c:53da prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:e8:96:56 txqueuelen 1000 (Ethernet)
    RX packets 6829037 bytes 9828791805 (9.8 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 533282 bytes 101254309 (101.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 895 bytes 65410 (65.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 895 bytes 65410 (65.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@hispark:/home/openharmony#

```

- 点击Windows的此电脑，鼠标右键，选择映射网络驱动器

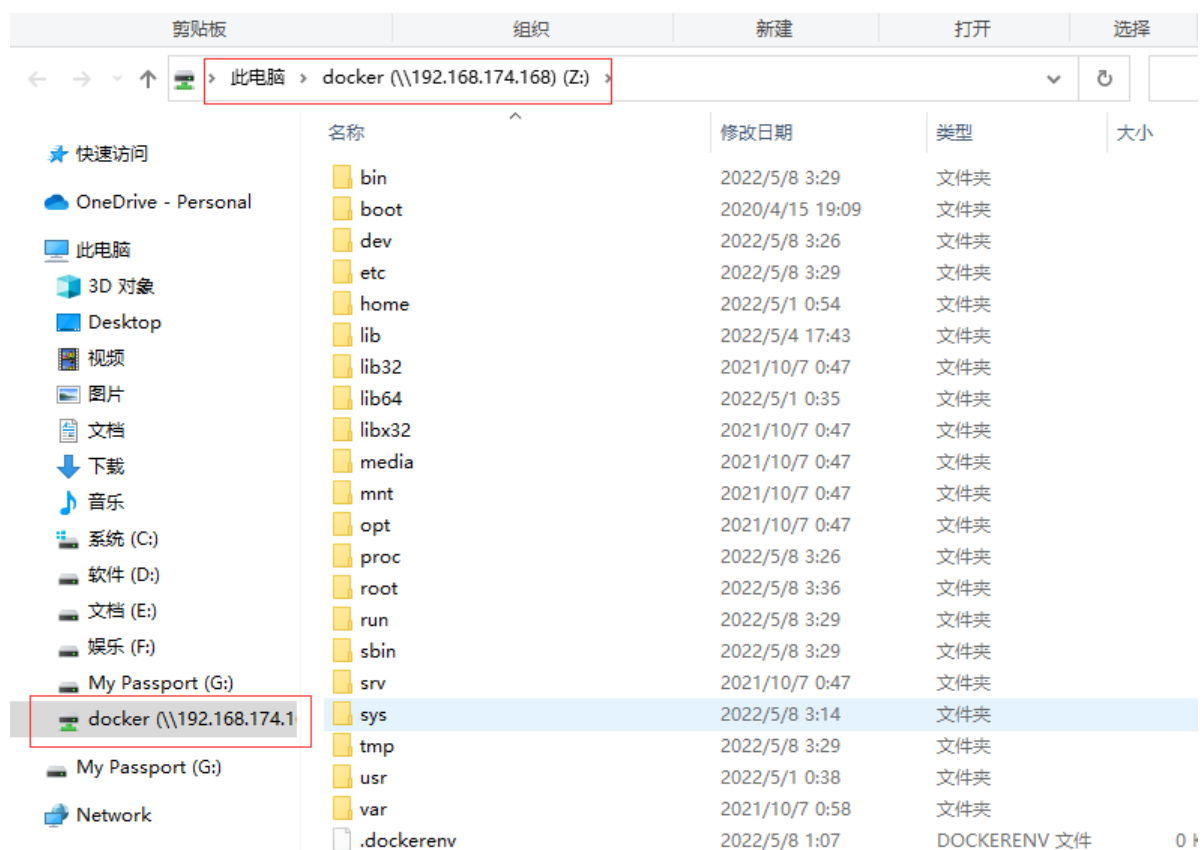


- 输入\\Ubuntu的IP地址\\docker，然后点击完成，输入账号(root)和初始密码(123456)

```
\\192.168.174.168\\docker
```

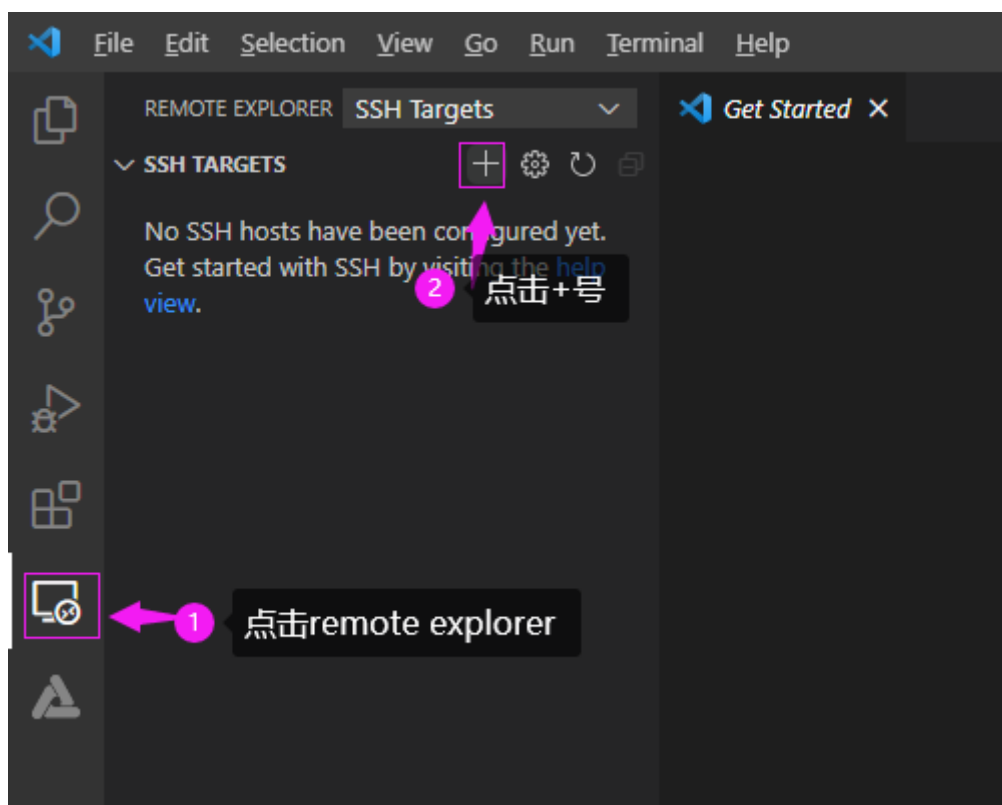


- 这样docker的根目录就能够在Windows的磁盘下面显示了。这样您就可以方便的进行Windows和Ubuntu之间的文件共享了



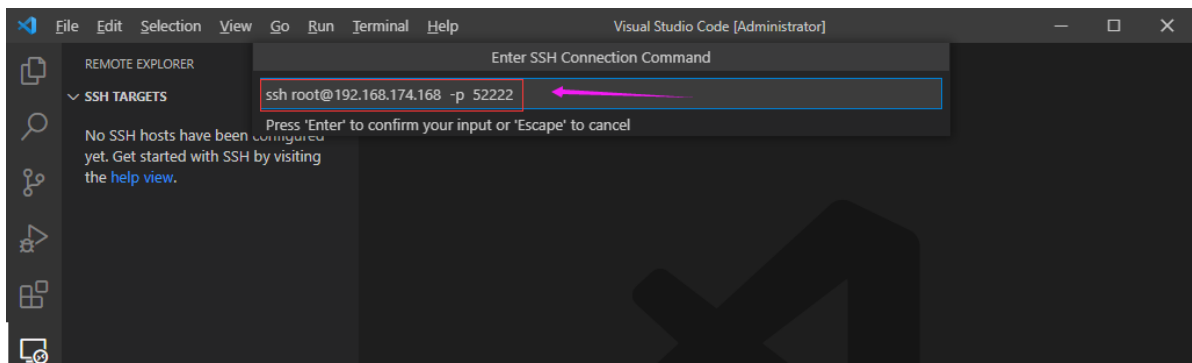
## 步骤8：使用VSCode,通过ssh与Ubuntu建立链接

- 打开VSCode, 点击remote explorer, 然后再点击ssh targets右边的加号

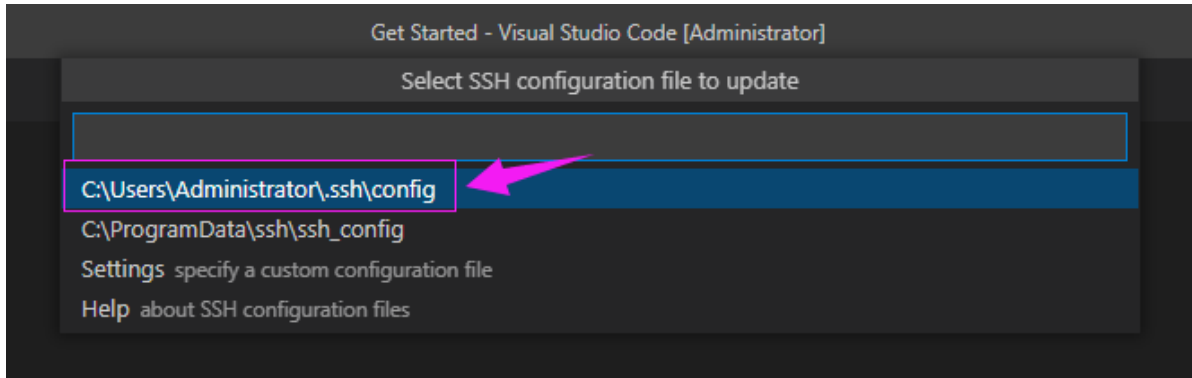


- 输入如下的指令, 然后敲回车, 默认密码是: **123456**

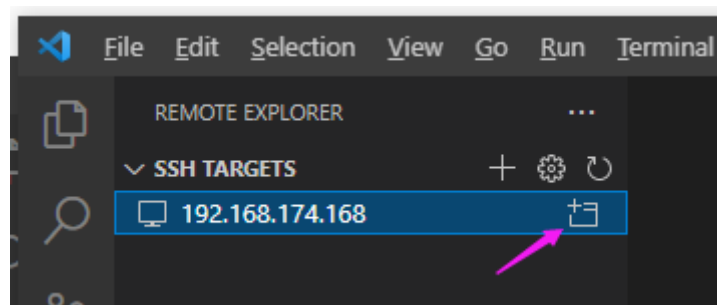
```
ssh root@ubuntu的ip地址 -p 52222
```



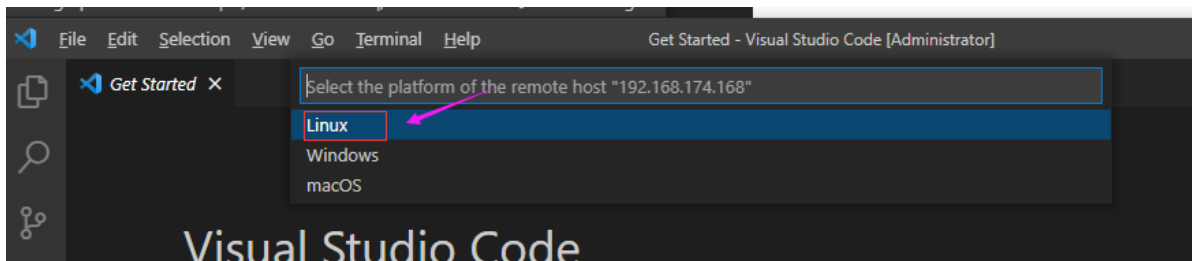
- 选择第一个config



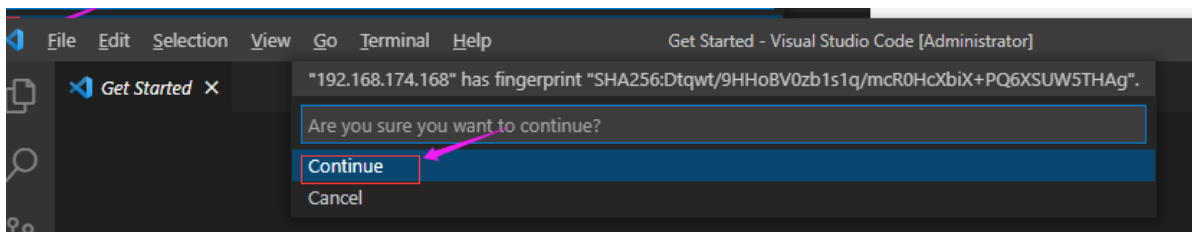
- 点击SSH TARGETS下面IP地址 右边的+号



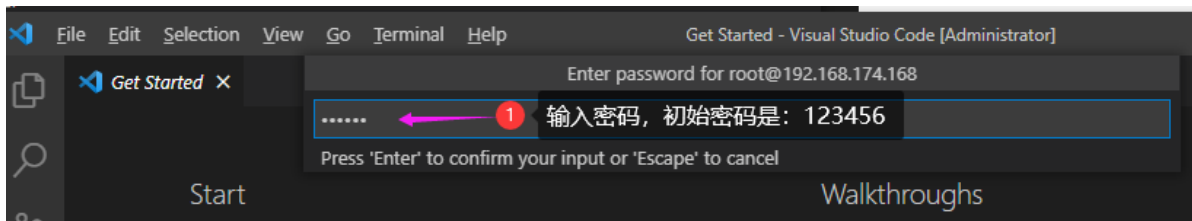
- VSCode上方会弹出下图所示的对话框，请选择Linux。



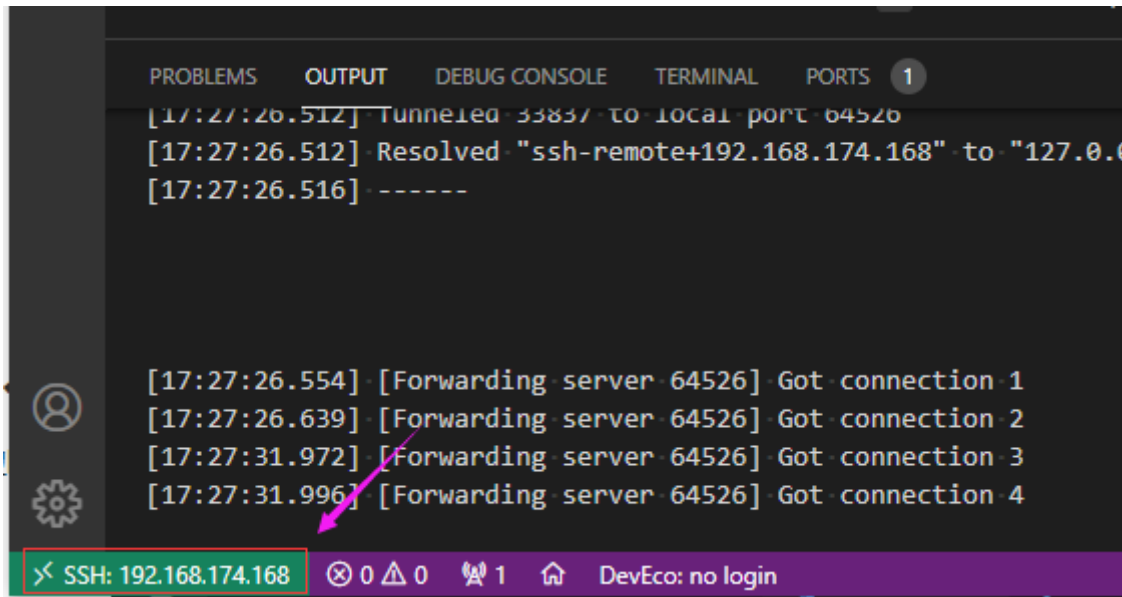
- VSCode上方会弹出对话框，点击continue。



- 输入密码，初始密码是：123456

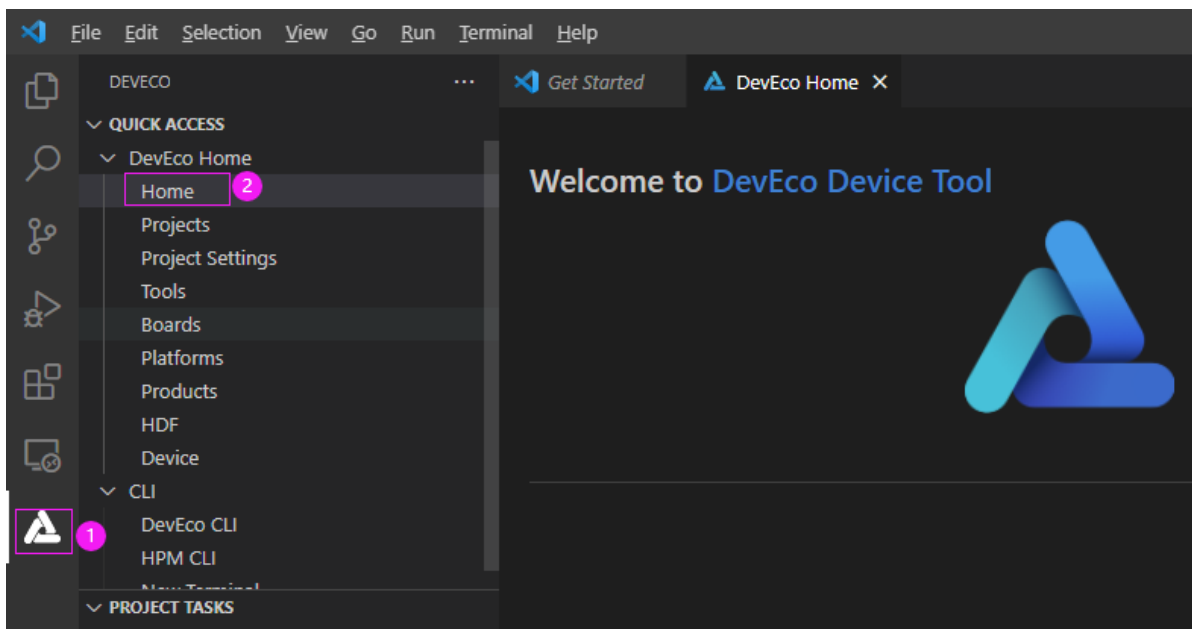


- 当左下角显示为SSH:IP地址, 且为绿色的时候, 说明已经和Ubuntu建立链接了。

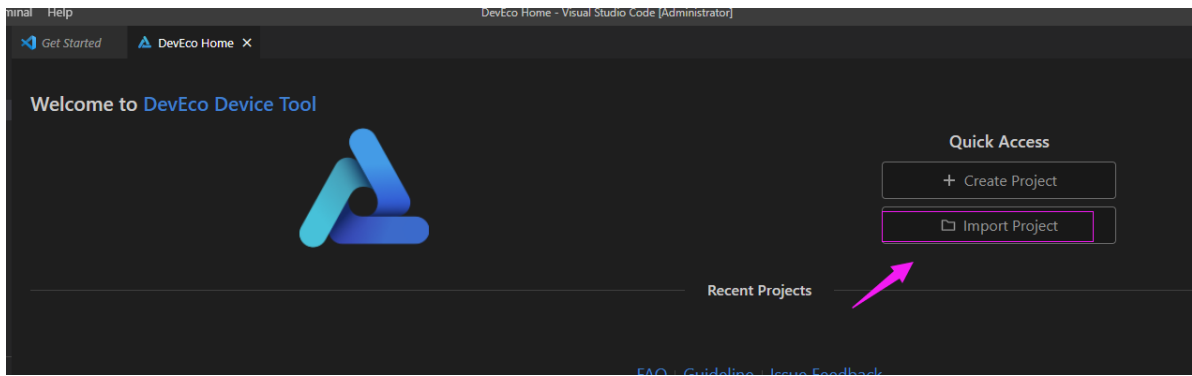


## 步骤9: 导入Hi3516DV300的工程

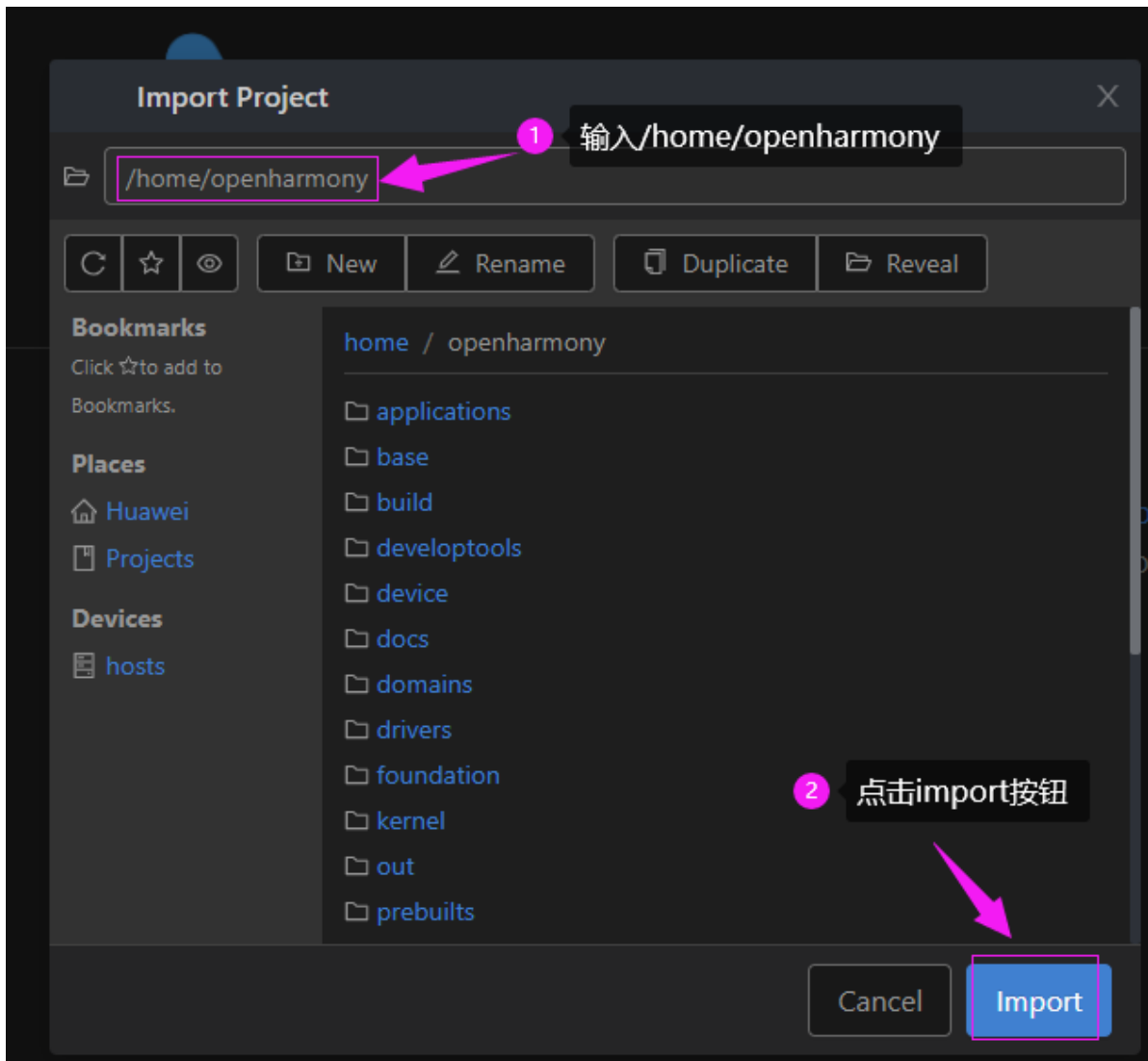
- 首先点击左边的DevEco图标, 然后点击DevEco Home下面的Home。



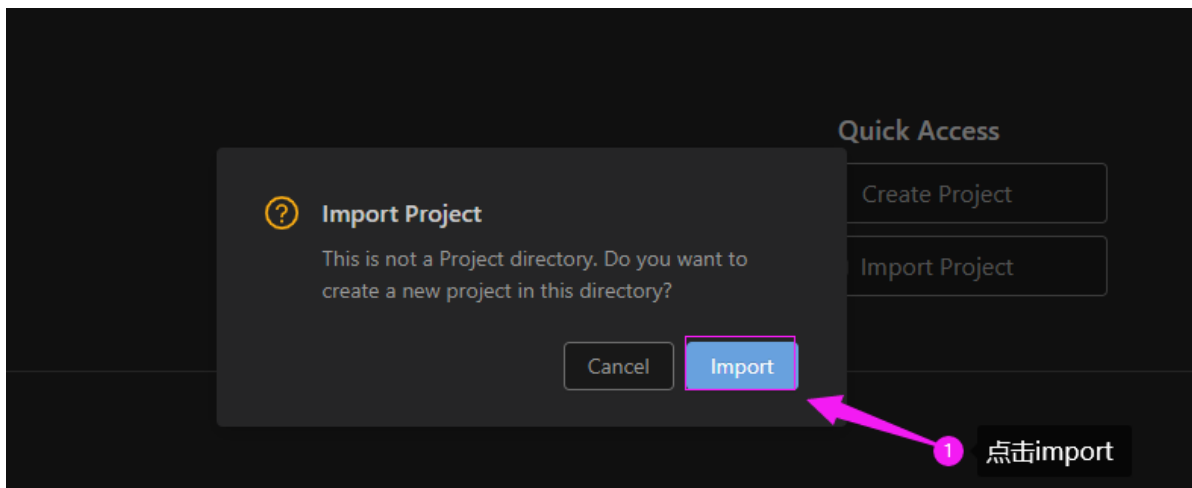
- 点击import Project按钮



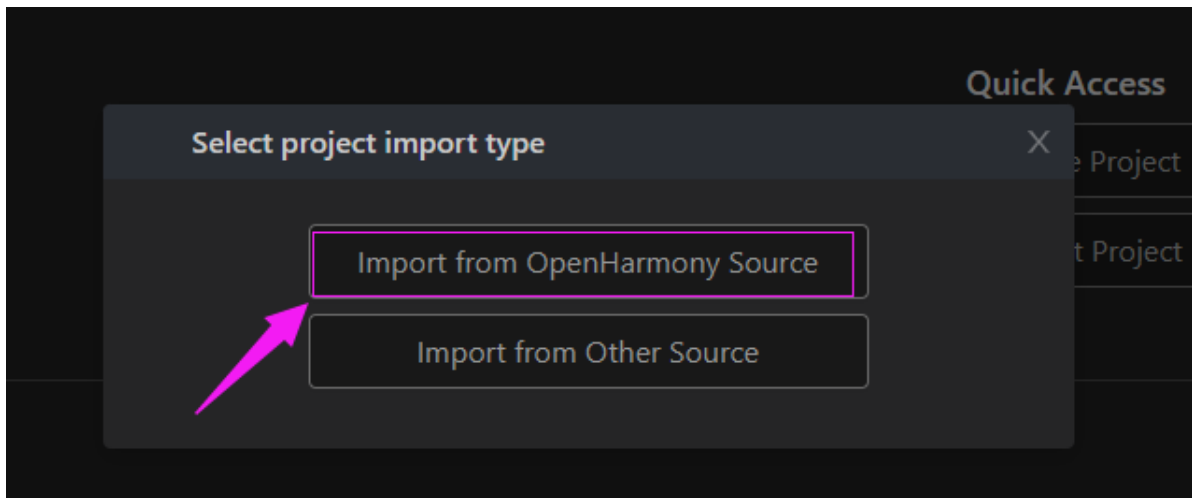
- 然后再import Project下面的路径，输入 /home/openharmony，然后再点击 import按钮。



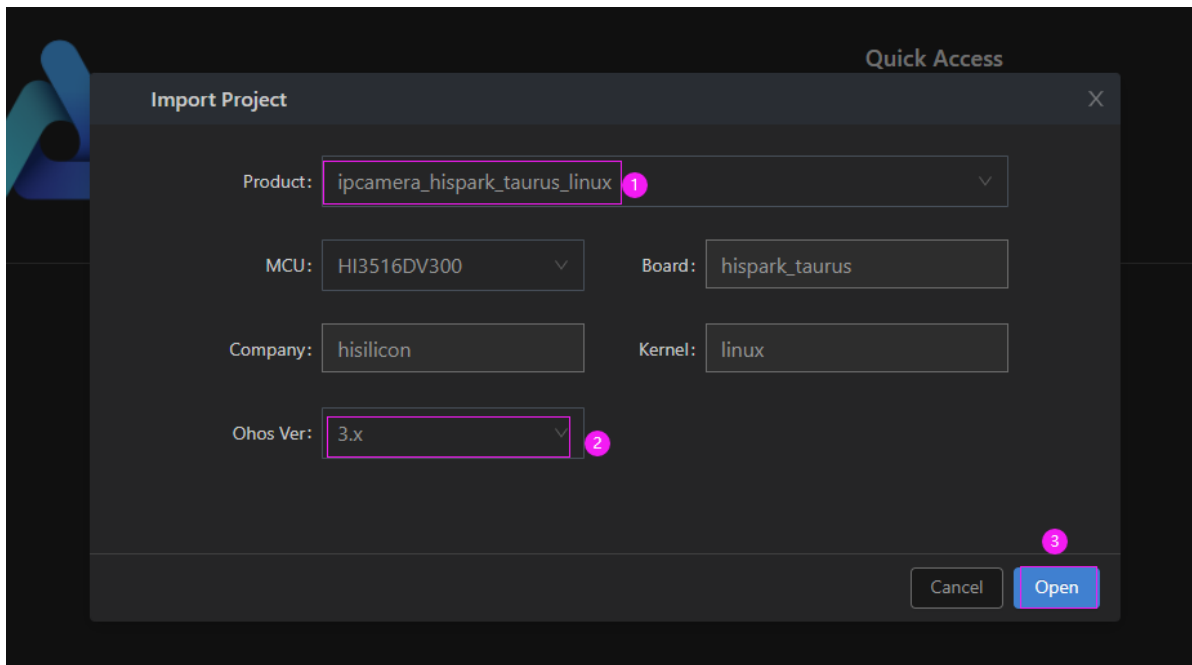
- 弹出对话框之后，再点击import。



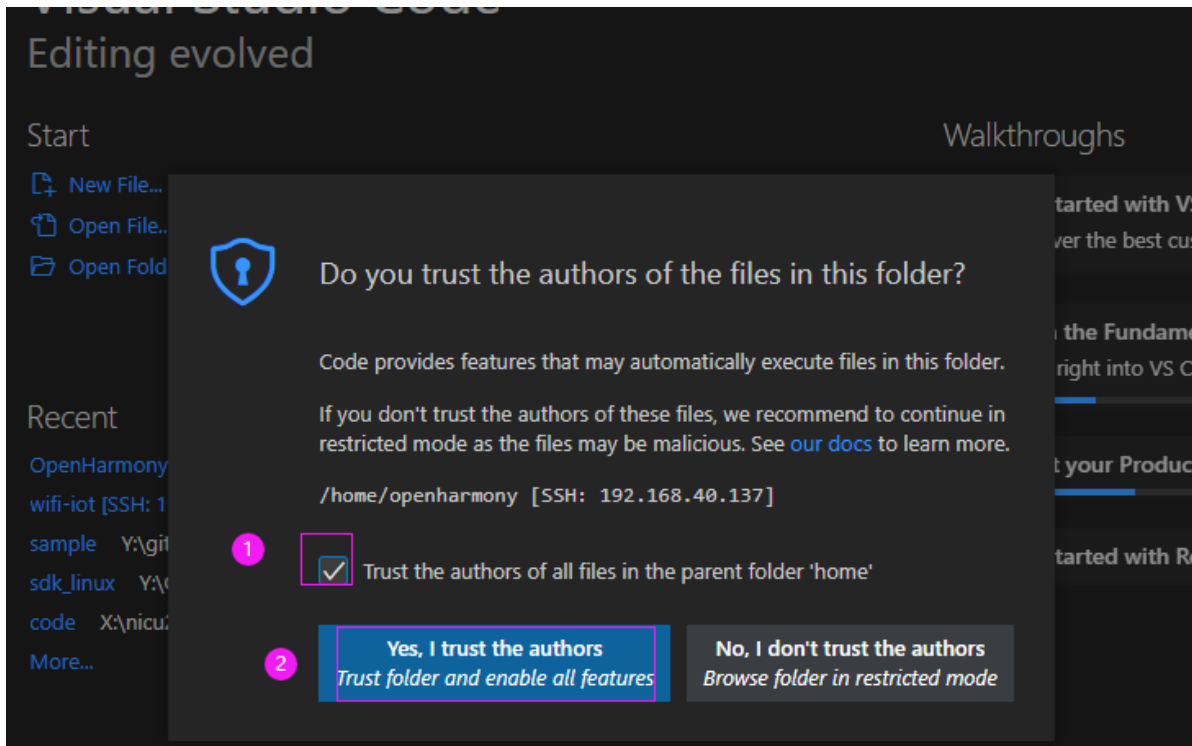
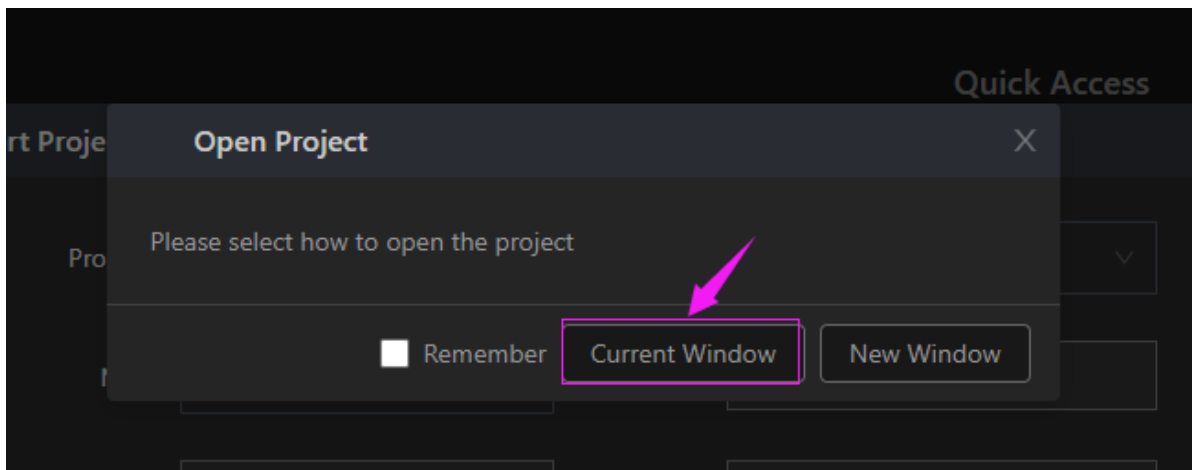
- 然后再点击import from OpenHarmony Source



- 然后选择Product为: ipcamera\_hispark\_taurus\_linux, 然后ohos Ver选择为3.0。



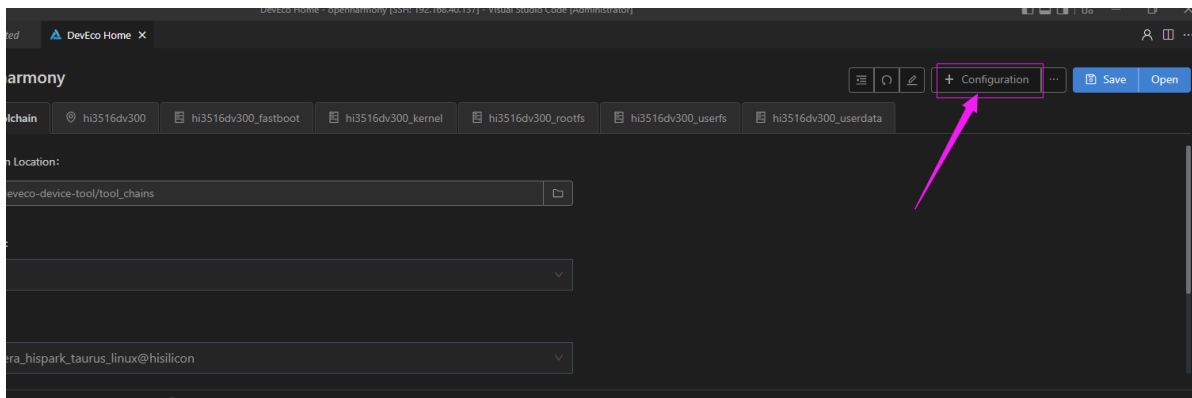
- 点击Current Windows



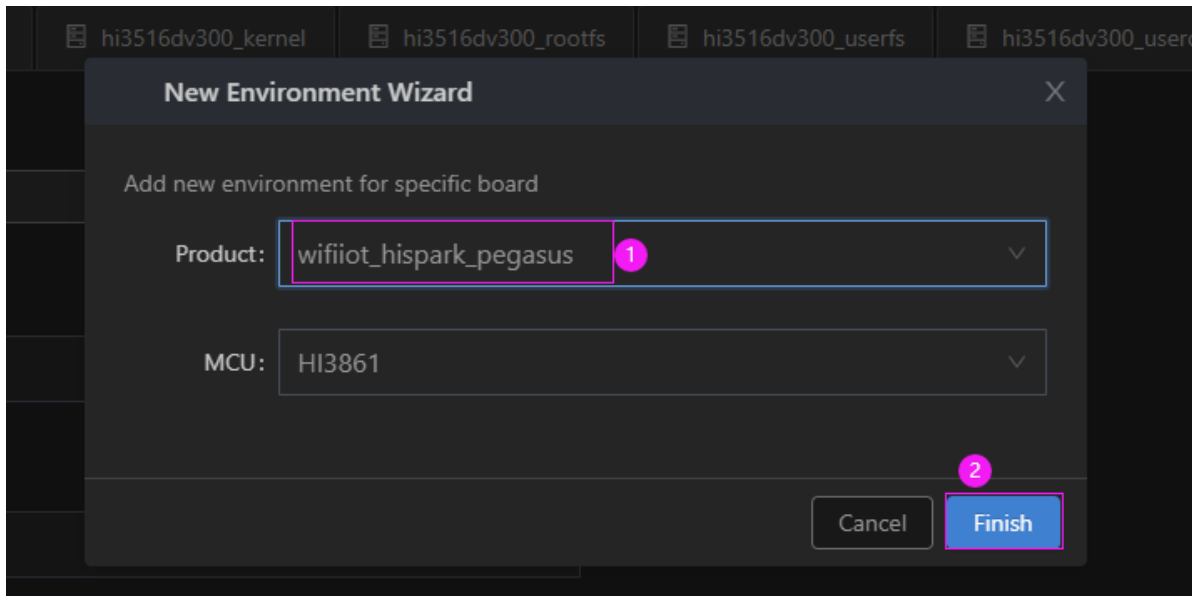
- 点击左边的Deveco图标，然后选择Project Settings，然后点击toolchain，确保下面的setup上面是绿色的对钩，说明环境已经OK了，如果是红色的X，说明环境还有缺失，需要点击setup，下载编译时所需的工具。
- 最后点击build按钮，进行代码的编译。



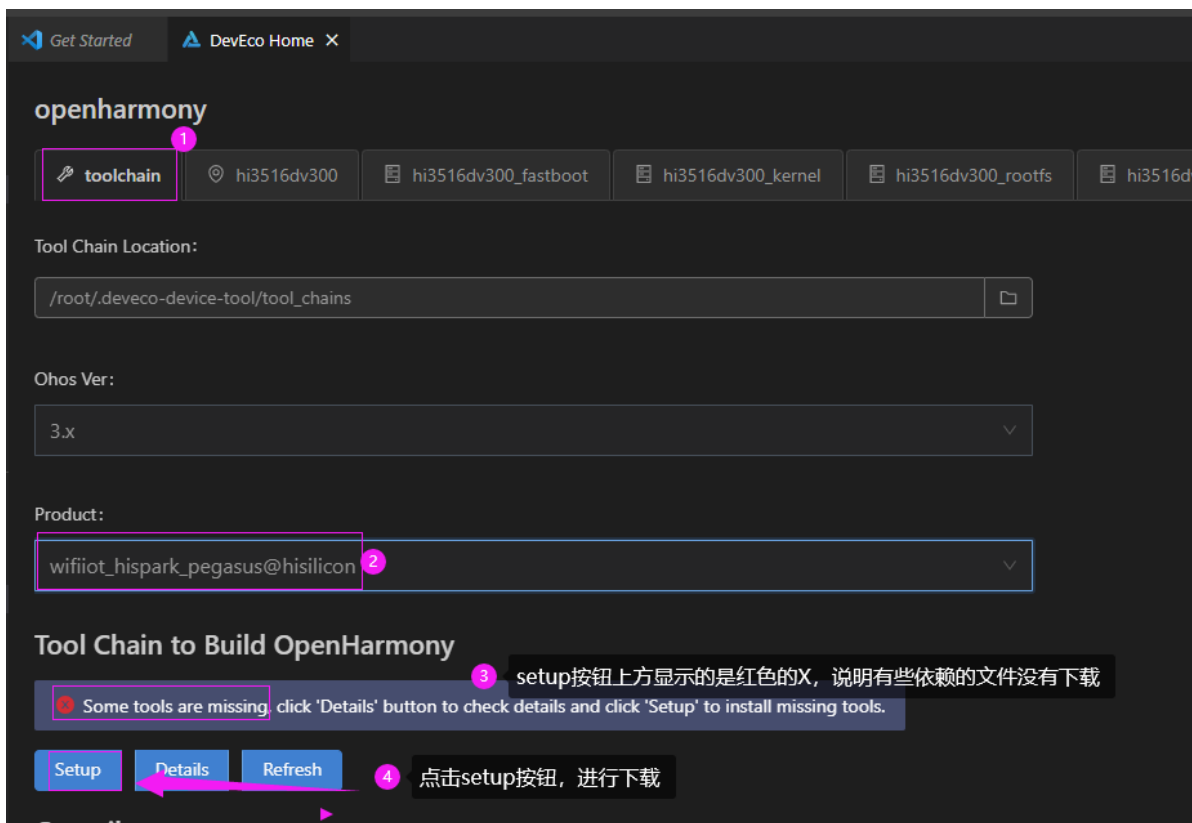




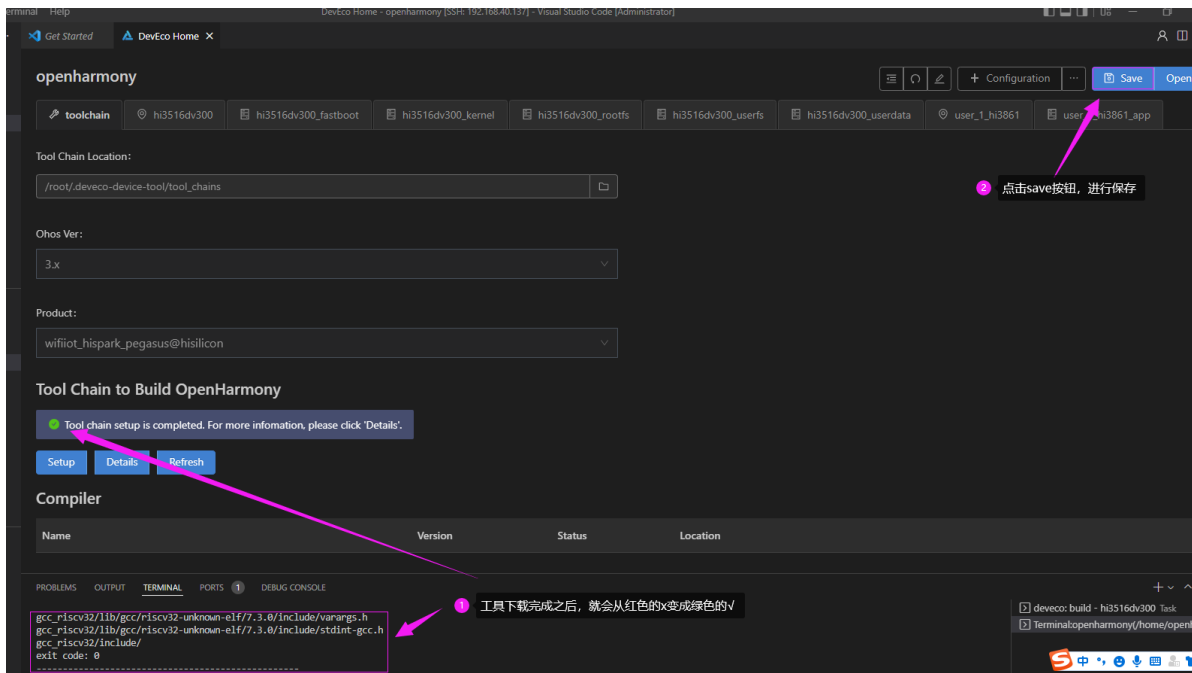
- product选择 wifiiot\_hispark\_pegasus，然后再点击finish。



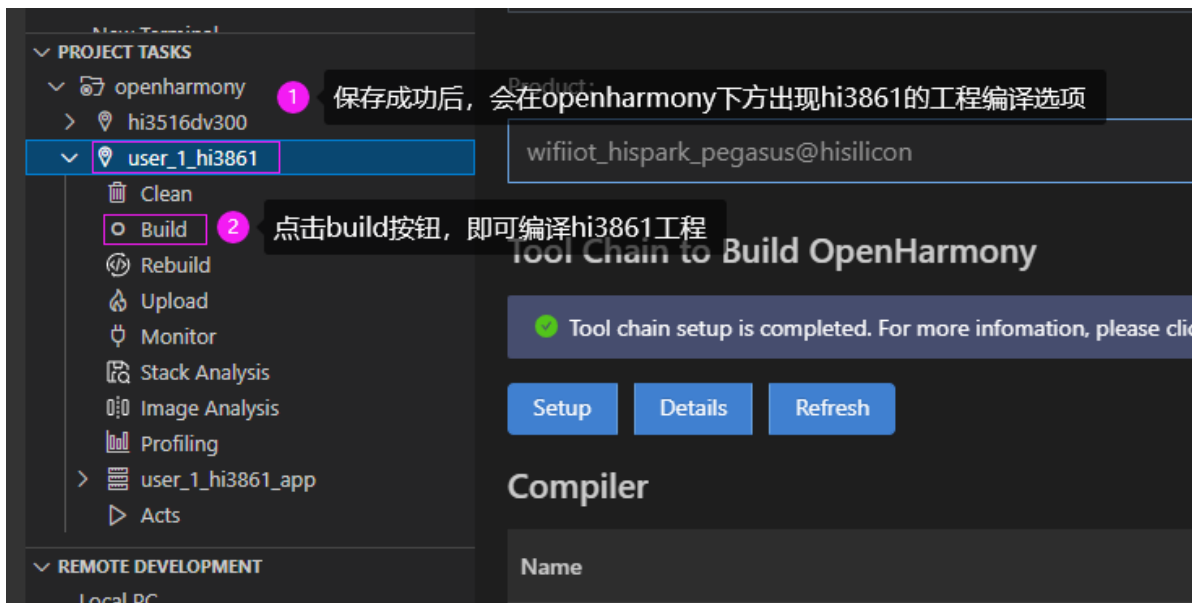
- 点击toolchain，下拉Product，选择wifiiot\_hispark\_pegasus@hisilicon，我们可以看到，setup按钮上面显示的是红色的x，说明编译Hi3861时依赖的一些工具是缺失的，所以需要点击setup按钮，进行下载。



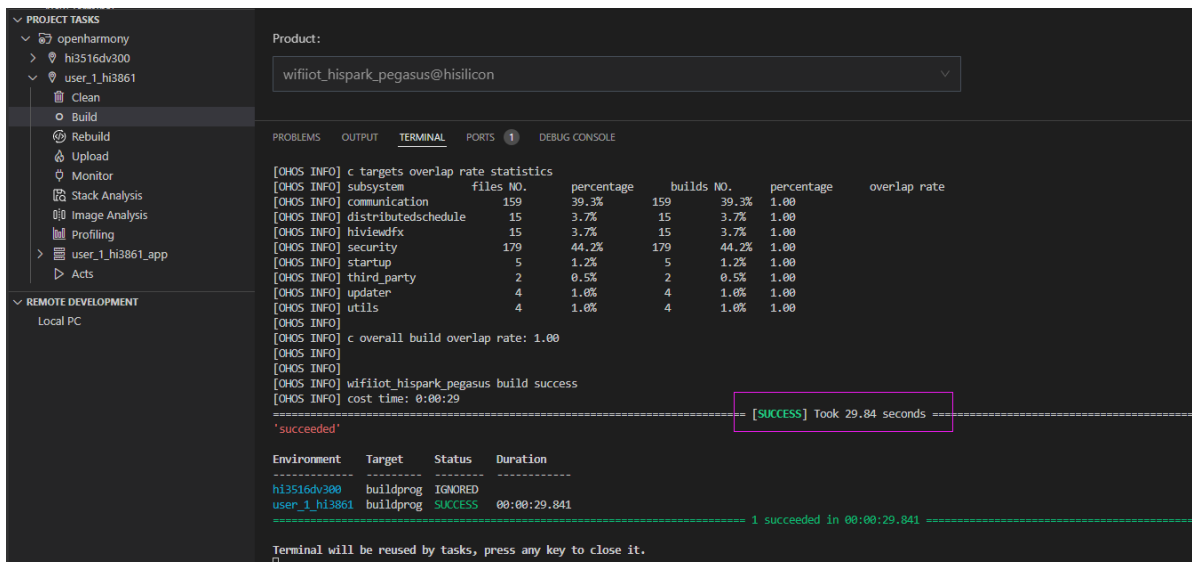
- 工具下载完成之后，红色的×就会变成绿色的√，然后点击右上角的保存按钮。



- 保存成功后，在会PROJET TASKS下方出现一个Hi3861工程的编译选项，点击build按钮，即可编译Hi3861的代码。



- 当出现success的字样，说明hi3861的工程就编译成功了。



- 关于Hi3861的烧录过程请参考《[Hi3861V100开发板烧录](#)》

## 附录：

- 文件拷贝

# 1、主机拷贝文件到docker编译环境里：

```
sudo docker cp 源文件 openharmony:/目标文件
```

# 参数解析：

# 源文件：主机上的，可为文件或者目录

# 目标文件：docker编译环境里的，通常为目录，表示将文件拷贝到该目录下

# 2、docker编译环境拷贝文件到主机：

```
sudo docker cp openharmony:/源文件 目标文件
```

# 源文件：docker编译环境里的，可为文件或者目录

# 目标文件：主机上的，通常为目录，表示将文件拷贝到该目录下

- 删除docker编译环境【谨慎操作，不可恢复】

# 查看当前运行的docker实例状态

```
sudo docker ps -a
```

# 在上一条指显示结果列表中，查看openharmony的STATUS

# 如为 up，则需要执行下面这条指令，停止其运行

# 如为 Exited，则跳过下面这条指令

```
sudo docker stop openharmony
```

# 删除

```
sudo docker rm openharmony
```

## 提醒：

**删除前，请确保该运行环境内的有效数据都已拷贝到主机上**

**删除后，该运行环境内的所有数据将被移除，不可恢复**